





АnalogWriteQ көмегімен ендік-импульстік модуляциялау






Осыған дейін Arduino-ның цифрлық контактілерін бақылауды қарастырдық. Олар жарықдиодтарын қайта қосу, реле және тұрақты ток қозғалтқыштарын басқару үшін өте ыңғайлы. Бірақ, 0 және 5 В-тан басқа кернеуді шығару керек болса, не істейді. Бір ғана Arduino Uno платасының контактілері арқылы оны жүзеге асыру мүмкін емес. Цифрлықаналогтық түрлендіргішті қолдану немесе Arduino Due пайдалану, болмаса сыртқы микросхемаларды қосу керек болады.



Дегенмен, цифрлық контактілердегі аналогтық мәндерді генерациялауды ендік-импульстік модуляция (ЕИМ) көмегімен имитациялауға болады. Arduino-ның кейбір контактілері үшін `analogWrite()` командасы арқылы ЕИМ сигналын құруға болады. Arduino платасында белгілі бір перифериялық құрылғыларға ЕИМ сигналын шығаратын контактілер ~ символымен белгіленген. Arduino UNO-да 3, 5, 6, 9, 10, 11 контактілері ЕИМ сигналын шығара алады. Arduino Uno платасында `analogWrite ()` командасын 4-дәрістегі 3-суретте көрсетілген схеманың көмегімен тексеруге болады.

Егер Arduino-ның 9 контактісіндегі кернеуді азайтса, жарықдиодының жарықтығы азаюы керек, себебі, ол арқылы өтетін ток азаяды. Бұл эффектіні ЕИМ және `analogWrite ()` командасының көмегімен алуға болады.



Analogwrite () функциясы екі аргументке ие: контакт нөмірі және осы контактіге орнатылатын, 0-ден 255-ке дейінгі аралықтағы 8 разрядты мән.

3 листингте жарықдиодының жарықтығын бірқалыпты басқару үшін 9 контактіде ЕИМ сигналын генерациялау бағдарламасының коды берілген.

Листинг 3. Жарықдиодының жарықтылығын бірқалыпты өзгерту


```
const int LED=9; // жарықдиоды контактісі нөмірінің константасы

void setup()
{
  pinMode (LED, OUTPUT); // жарықдиоды контактісін контакт ретінде конфигурациялаймыз
}

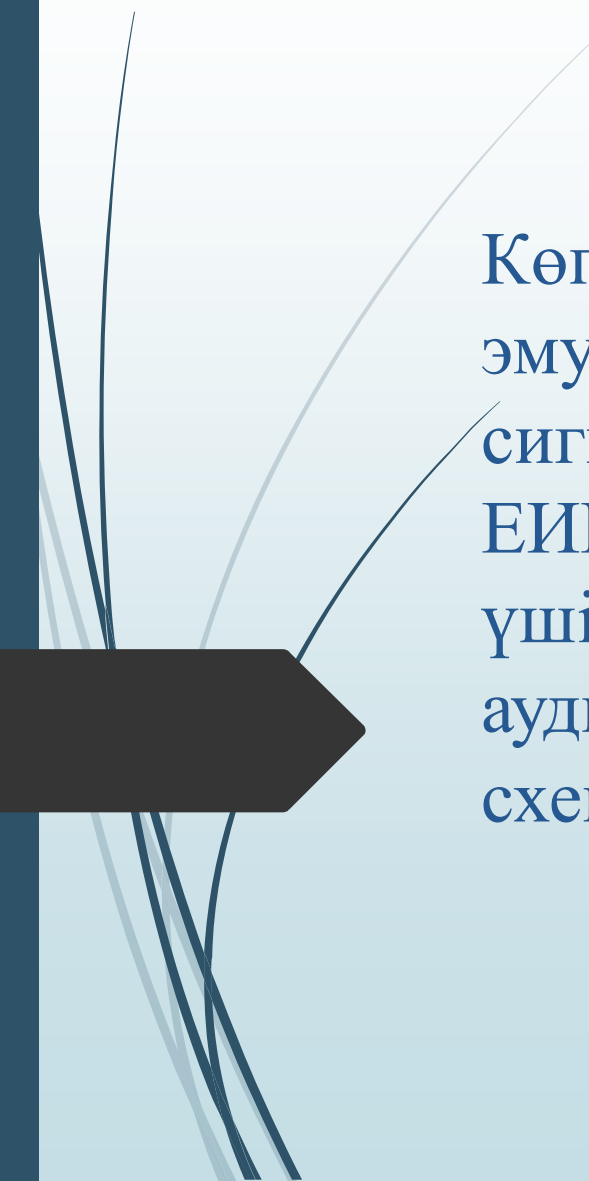
void loop()
{
  for (int i=0; i<256; i++)
  {
    analogWrite(LED, i);
    delay(10);
  }
  for (int i=255; i>=0; i--)
  {
    analogWrite(LED, i);
    delay(10);
  }
}

const int LED=9; //


void setup()
{
  pinMode (LED, OUTPUT); //
}
```



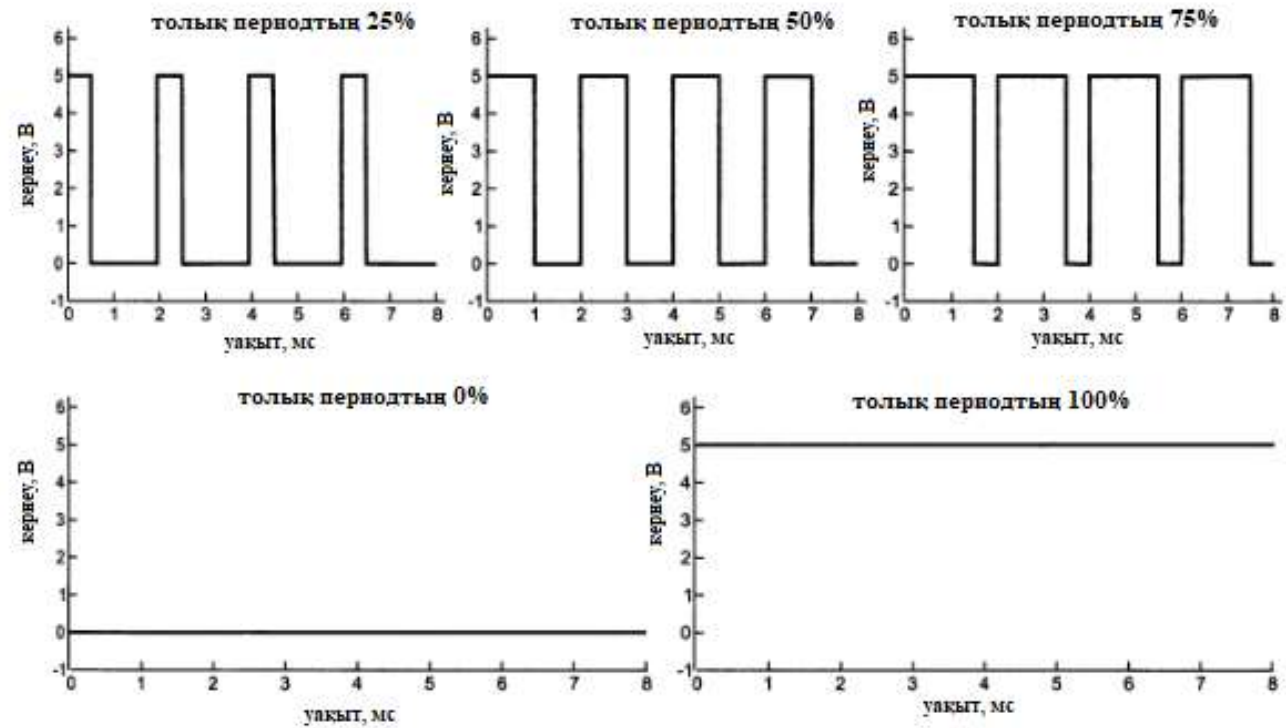
3 листинг жүктелген кезде жарықдиоды не орындайды?
for - дың бір циклінде жарықдиодтың жарықтылығының күңгірттен жарыққа, содан кейін екінші циклде жарықтан күңгіртке ауысатынын бақылауға болады. Мұның барлығы loop () негізгі циклінде шексіздікке дейін орындала береді. Екі цикл арасындағы айырмашылыққа назар аударайық. Бірінші циклде $i++$ өрнегі $i=i+1$ кодының қысқартылуы болып табылады. Сол сияқты $i--$ жазбасы $i=i-1$ кодына сәйкес. Бірінші цикл жарықдиодын максималды жарықтылыққа дейін бірқалыпты жағады, екінші цикл бірқалыпты сөндіреді.



Көптеген жағдайларда ЕИМ аналогтық шығысты эмуляциялауға жарайды, бірақ, бұрмаланбаған аналогтық сигнал қажет болған кезде, бұл нұсқа жарамайды. Мысалы, ЕИМ тұрақты ток қозғалтқышының жылдамдығын реттеу үшін өте жақсы (келесі дәрістерде қарастырылады), бірақ аудиоколонкаларды басқаруға жарамайды (қосымша сыртқы схема қажет болады).



Барлығын түсіну үшін, ЕИМ қалай жұмыс істейтінін 1 суретте берілген графиктер арқылы қарастырайық. ЕИМ импульстардың тікбұрышты тізбегіндегі жұмыс циклының (скважность) (периодтың импульстің ұзақтығына қатынасы) өзгеруі болып табылады. Жұмыс циклын қайталаудың барлық периодында, тікбұрышты импульс HIGH деңгейіне ие кездегі уақыт проценті ретінде түсіндіруге болады. 50% жұмыс деңгейі - периодтың жартысында сигнал жоғары деңгейге, ал жартысында – төменге ие екенін білдіреді.



1 сурет. Түрлі жұмыс цикліндегі ЕИМ сигналы

`analogwrite ()` функциясы оған берілген мәнге байланысты тікбұрышты импульстар тізбегінің жұмыс циклын орнатады:

- ◆ `analogwrite ()` аргументінің нөлге тең мәні 0% -ға тең жұмыс циклын орнатады (әрқашан LOW);
- ◆ 255 мәні - жұмыс циклы 100% (әрқашан HIGH);
- ◆ 127 мәні 50% жұмыс циклына сәйкес келеді (уақыттың жартысы HIGH, уақыттың жартысы LOW).

1 суреттегі графиктерден 25% жұмыс циклы бар сигнал үшін HIGH мәні кезеңнің төрттен бір бөлігінде, ал уақыттың қалған 75%-ы LOW деңгейге орнатылғанын көруге болады.